IronMask: Modular Architecture for Protecting Deep Face Template

Sunpill Kim[1] Yunseong Jeong[2] Jinsu Kim[3] Jungkon Kim[3] Hyung Tae Lee[2] Jae Hong Seo[1]*

[1]Department of Mathematics & Research Institute for Natural Sciences, Hanyang University
[2]Division of Computer Science and Engineering, College of Engineering, Jeonbuk National University
[3]Security Team, Samsung Research, Samsung Electronics

Abstract

Convolutional neural networks have made remarkable progress in the face recognition field. The more the technology of face recognition advances, the greater discriminative features into a face template. However, this increases the threat to user privacy in case the template is exposed.

In this paper, we present a modular architecture for face template protection, called IronMask, that can be combined with any face recognition system using angular distance metric. We circumvent the need for binarization, which is the main cause of performance degradation in most existing face template protections, by proposing a new real-valued error-correcting-code that is compatible with real-valued templates and can therefore, minimize performance degradation. We evaluate the efficacy of IronMask by extensive experiments on two face recognitions, ArcFace and Cos-Face with three datasets, CMU-Multi-PIE, FEI, and Color-FERET. According to our experimental results, IronMask achieves a true accept rate (TAR) of 99.79% at a false accept rate (FAR) of 0.0005% when combined with ArcFace, and 95.78% TAR at 0% FAR with CosFace, while providing at least 115-bit security against known attacks.

1. Introduction

Biometric-based authentication has long been the subject of intensive research, and deep convolutional neural networks (CNNs) have allowed significant recent advances in recognition. The success of the CNN architecture comes from its ability to extract discriminative features from user biometric traits, such as facial images. Within face recognition systems, user face images are well-compressed into deep face templates. Therefore, the exposure of user face templates constitutes a severe threat to user privacy as well as to system security. For instance, face images can be reconstructed from the corresponding templates [20] and this

is a direct threat to user privacy. In addition, the reconstructed face image can be used to gain unauthorized access to the system. A high number of attacks have already been reported, which shows how risky the leakage of templates is [20, 4, 10, 9, 3].

The goal of face template protection is to satisfy *irreversibility*, *revocability*, and *unlinkability* properties without significant degradation in face recognition performance. In terms of security, in the event a protected face template is leaked to an adversary, reconstructing the original face template from the compromised template should be computationally infeasible. Thus, the compromised template must be revoked and generation of a new protected face template should be feasible. In addition, two or more protected templates created from the same user biometric traits should hide relations between them. Furthermore, the template protection scheme is expected to maintain high matching performance with high intra-user compactness and inter-user discrepancy. Achieving both high template security and high match performance at the same time is a fundamental challenge in the design of face template protection schemes.

Most of recent research on CNN-based face template protection can be categorized into two groups, *training-in-enrollment* and *training-before-enrollment*. The former approach [23, 12, 13] requires face images of the target user to train the CNN, which maps from face images into binary code during enrollment; at this stage, the resulting binary codes are randomly generated and so have no correlation with the original face images. The hashed value of the resulting binary code is stored as a protected template of the target user. This approach is incompatible with dynamic enrollment systems because, once the initial enrollment has been completed, the CNN would need to be retrained from scratch whenever a new user is added to the system. The latter approach [31, 21] was proposed to mitigate the effect of the above problem. In this case, the CNN can be pre-trained before enrollment, and performance degradation is minimized using additional methods, e.g., neural network

---

*Corresponding author: J. H. Seo (e-mail: jaehongseo@hanyang.ac.kr)
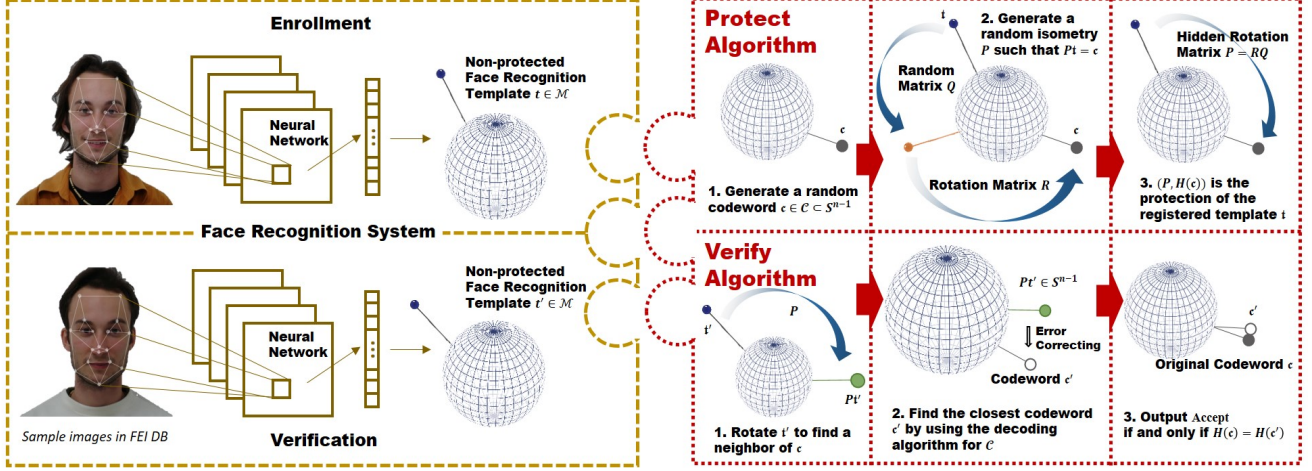
Figure 1: IronMask is a template protecting modular architecture, which can be combined with any face recognition systems using angular distance metric.

based error-correction [31] and significant feature extraction [21]. However, compared to the training-in-enrollment approach, these algorithms suffer from low matching accuracy. The common feature of both approaches is the use of a binary code either as an intermediate step or as the final protected template. This artificial binarization makes template-protected face recognition systems quite different from state-of-the-art (non-protected) face recognition systems in terms of both architecture and/or matching performances.

1.1. Our Contributions

We present a novel design architecture that maintains the advantages of state-of-the-art (non-protected) biometric recognition as much as possible, while adding high template security. To this end, we propose a modular architecture, called IronMask, that provides template protection to any (non-protected) face recognition system using angular distance metric. A pictorial explanation of the approach is provided in Fig. 1. By combining IronMask with well-known face recognition systems, we achieve the best matching performance in the training-before-enrollment category. Our performance results are even comparable with existing results in the training-in-enrollment category.

In this paper, we consider an arbitrary biometric recognition system, which measures the closeness of two templates by angular distance metric (or cosine similarity). Two biometric templates $\mathbf{t}_1, \mathbf{t}_2$ are considered as vectors on the unit n-sphere S^{n-1}, and the distance between them is defined as the angle $\cos^{-1}(\langle \mathbf{t}_1, \mathbf{t}_2 \rangle)$ between them, where $\langle \mathbf{t}_1, \mathbf{t}_2 \rangle$ denotes the standard inner product of $\mathbf{t}_1$ and $\mathbf{t}_2$. It is worth noting that the angular distance [11, 18, 19, 25, 36, 17, 35, 37, 5, 39] has been widely used along with the Euclidean distance in many recent successful face recognition systems

to measure closeness between two templates.

In developing IronMask, we devise a new real-valued error-correcting-code (ECC) that is compatible with the angular distance metric and enables to use real-valued codes instead of binary codes, unlike prior works. Hence, given an original template $\mathbf{t} \in S^{n-1}$, IronMask generates a real-valued codeword $\mathbf{c}$ and a linear transformation $\mathbf{P}$ mapping from $\mathbf{t}$ to $\mathbf{c}$, which has the form of an n-by-n matrix. The hashed value $\mathsf{H}(\mathbf{c})$ of the codeword and $\mathbf{P}$ are stored as the protected template, where H is a collision resistant hash function. If we set $\mathbf{P}$ to be an orthogonal matrix, then it becomes an *isometry* that preserves the inner product, so that any noisy template with distance d from $\mathbf{t}$ can be transformed via $\mathbf{P}$ into the one with the same distance from $\mathbf{c}$. Thus, no additional performance degradation is imposed by this transformation. Finally, with the help of the error-correcting capability of our ECC, we can recover the original codeword and check its originality with $\mathsf{H}(\mathbf{c})$.

If the protected template is leaked, the adversary gains only $\mathbf{P}$ and $\mathsf{H}(\mathbf{c})$. We inject sufficient randomness into $\mathbf{P}$ to hide information on the original template via the mathematical tool of choosing a uniform orthogonal matrix. Then, we show that the protected template satisfies the three required properties, irreversibility, revocability, and unlinkability under appropriate parameter sizes.

IronMask can be combined with arbitrary face recognition systems using angular distance metrics. To demonstrate the efficacy of IronMask, we apply IronMask to two well-known face recognition systems, ArcFace [5] and Cos-Face [37], with three widely used datasets, CMU Multi-PIE [28], FEI [33], and Color-FERET [24]. Here, ArcFace and CosFace are one of the state-of-the-art face recognition systems. Our experiments show that IronMask with Arc-Face achieves 98.76% true accept rate (TAR) at 0% false ac-

cept rate (FAR), and with CosFace achieves 81.13% TAR at 0% FAR. Through additional processes involving multiple face images as input of IronMask, we improve the matching performance up to 99.79% TAR at 0.0005% FAR for ArcFace and 95.78% TAR at 0% FAR for CosFace.

1.2. Related Works

Many biometric template protection techniques have been proposed in the literature. Secure sketch [6], fuzzy commitment [14], and fuzzy vault [15] are biometric cryptographic schemes that not only enable rigorous analysis on template security but also guarantee high template security. These cryptographic solutions employ ECCs and cryptographic hash functions as main building blocks. In the cryptographic approach, an original template is mapped onto a random codeword, and its hash value is subsequently stored as the protected template. These schemes vary according to the metric of the space containing the templates. Many face template protection proposals [30, 1, 38, 31, 21] follow this approach. However, this requires to use binary codes or points, which leads to loss of discriminatory information on the original template and degradation of matching performance. Biohashing [32], cancelable biometric [26], and robust hashing [34] are feature transformation approaches that transform the original template into an element in a new domain using a non-invertible transformation and salting. A hybrid approach for face template protection that combines the cryptographic approach with the feature transformation approach was proposed in [8]. Local region based hashing was suggested as a way to extract discriminative features from face images [22]; however, the feature extraction consists of quantization and cryptographic hashing, so that this algorithm still suffers from low matching accuracy in the same way as the cryptographic approach. To improve matching performance as well as security, several studies have proposed CNN-based approaches [23, 12, 13], which minimize intra-user variability and maximize inter-user variability using neural networks. These works [23, 12, 13] essentially require one or multiple face image(s) of the target user when training the neural networks, and this requirement is impractical in some applications such as dynamic enrollment systems.

There is another approach [2, 7] based on fully homomorphic encryption that provides a strong security against attacks using biometric distribution. However, unlike aforementioned works and ours, its security is guaranteed only under the additional assumption that the decryption key of fully homomorphic encryption is kept secret.

2. Modular Architecture for Protecting Templates

The more the technology of face recognition advances, the greater the number of discriminative features that are well-compressed into a face template. Ironically, this phenomenon can be demonstrated by applying previously proposed face image reconstruction attacks to the state-of-the-art face recognition system. For example, we apply the face image reconstruction attack that was originally proposed for attacking Facenet [27], to ArcFace [5], which is one of the state-of-the-art face recognition system. It shows that the success probability of the attack against ArcFace is higher than that against FaceNet. Refer to Appendix A of the full version [16] of this paper for the details.

In this section, we propose a modular architecture for protecting angular distance based templates including those of ArcFace. More precisely, we present a transformation, called IronMask, which transforms a (non-protected) biometric recognition system using angular distance metric into a template-protected one. Biometric recognition systems can be classified into 'identification' and 'verification' according to usage scenarios. The goal of the identification is to identify a given template among templates in the repository. On the other hand, the verification is equivalent to the authentication. For the sake of simplicity, in this paper we focus on the verification, but note that the proposed protection method also works for the identification. IronMask is a transformation of the verification system and consists of two algorithms, template protection algorithm Protect and verification algorithm Verify. The Protect algorithm takes an original template as an input and returns a protected template. Given a template $\mathbf{t}$ and a protected template $\mathbf{pt}$, intuitively, it is non-trivial to check the closeness between $\mathbf{t}$ and $\mathbf{pt}$, so that a specialized verification algorithm Verify is necessary in our approach. The Verify algorithm takes a non-protected template $\mathbf{t}'$ and a protected template $\mathbf{pt}$ where $\mathbf{pt}$ is the output of the Protect algorithm with input $\mathbf{t}$, and then it outputs 1 (Accept) or 0 (Reject) according to the closeness between $\mathbf{t}$ and $\mathbf{t}'$.

Mathematical Concepts. We introduce some mathematical concepts used in the paper. A cryptographic hash function, denoted by H, is a one-way function such as SHA-256 that given a hash value, it is computationally infeasible to find its preimage. An error correcting code (ECC) is a set of codewords, $\mathcal{C}$, in the metric space $\mathcal{M}$ with decoding function Decode such that, informally speaking, given an element in $\mathcal{M}$, it finds the closest codeword in $\mathcal{C}$. In this paper, we are interested in the case $\mathcal{M} = S^{n-1}$, the set of all unit vectors in $\mathbb{R}^n$. An orthogonal matrix is a square matrix whose columns and rows are orthonormal vectors. Throughout the paper, $O(n)$ denotes the set of $n \times n$ orthogonal matrices.

IronMask: Transformation for Template Protection. We observe that templates in our target recognition systems are vectors in the unit n-sphere S^{n-1} with angular distance. Here, our basic idea for transformation is to use an orthogo-

nal matrix, which is an isometry keeping angular distance between templates once transformed. Then, the orthogonal matrix becomes a part of protected template. To generate this special orthogonal matrix, we define an algorithm HRMG that takes two inputs $\mathbf{a}, \mathbf{b} \in S^{n-1}$ and returns a random orthogonal matrix $\mathbf{P}$ such that $\mathbf{b} = \mathbf{Pa}$. This algorithm will be specified in Section 2.2. In addition, we use a sub-algorithm, called USample, to describe an abstract construction of IronMask. The USample algorithm generates a uniform element from the set of codewords and the details will be described in Section 2.1. Now we present the abstract construction of IronMask in Algorithms 1 and 2.

Algorithm 1: Protect Algorithm

Input : Template $\mathbf{t} \in \mathcal{M}$
Output: Protected template $\mathbf{pt}$

1 $\mathbf{c} \leftarrow$ USample
2 $\mathbf{P} \leftarrow$ HRMG$(\mathbf{t}, \mathbf{c})$ such that $\mathbf{c} = \mathbf{Pt}$ and $r \leftarrow$ H$(\mathbf{c})$.
3 Output $\mathbf{pt} := (r, \mathbf{P})$.

Algorithm 2: Verify Algorithm

Input : Protected template $\mathbf{pt}$ and template
$\qquad$ $\mathbf{t}' \in \mathcal{M}$
Output: *Accept* or *Reject*

1 Parse $\mathbf{pt}$ as $(r, \mathbf{P})$.
2 $\mathbf{c}' \leftarrow$ Decode$(\mathbf{P} \cdot \mathbf{t}')$ and $r' \leftarrow$ H$(\mathbf{c}')$.
3 Output *Accept* if $r' = r$. Otherwise, output *Reject*.

By the definition of HRMG$(\mathbf{t}, \mathbf{c})$, $\mathbf{P}$ is an isometry transformation from the original template $\mathbf{t}$ to the codeword $\mathbf{c}$. That is, since the orthogonal matrix $\mathbf{P}$ preserves inner products, the angular distance between the original template $\mathbf{t}$ and a newly recognized template $\mathbf{t}'$ is equal to that between $\mathbf{c} = \mathbf{Pt}$ and $\mathbf{c}' = \mathbf{Pt}'$. Therefore, if both $\mathbf{t}$ and $\mathbf{t}'$ are created from the same user, then we expect that $\mathbf{c}$ and $\mathbf{c}'$ have a small angular distance, so that Decode$(\mathbf{c}')$ returns $\mathbf{c}$ with high probability where Decode is the decoding function of the employed ECC.

Our abstract construction of IronMask is quite simple, but its realization is challenging mainly due to the following reasons:

1. We need a suitable candidate for ECC over S^{n-1} with angular distance.
2. The HRMG algorithm should prevent from leaking an important information on the input template $\mathbf{t}$.

In the next subsections, we will resolve the above two issues to complete our IronMask construction.

2.1. Error Correcting Codes over S^{n-1} with Angular Distance

In this subsection, we design a new real-valued ECC, which is a special building block for IronMask and quite different from usual binary ECCs such as BCH codes. To clarify the design rationale, we first list requirements that ECCs for IronMask should satisfy.

- (Discriminative) All codewords are well-spread on S^{n-1}, so that any two of codewords are sufficiently far to each other. This property is necessary for high accuracy of IronMask.
- (Uniformly-Samplable) There exists an efficient algorithm USample generating a codeword with uniform distribution over the set of all codewords. This property will be used in the Protect algorithm of IronMask.
- (Efficiently-Decodable) There exists an efficient algorithm Decode that takes any vector in S^{n-1} as an input and returns a codeword with the shortest angular distance from the input vector. This property will be used in the Verify algorithm of IronMask.
- (Template-Protecting) The number of codewords is sufficiently large in order to prevent the brute-force attack, e.g., 2^{80} for 80 bits security.

Next, we devise a new ECC satisfying all the above four requirements. To this end, we first define a family of codeword sets $\{\mathcal{C}_\alpha\}_\alpha$.

Definition 1. *For any positive integer α, $\mathcal{C}_\alpha$ is defined as a set of all unit vectors whose entries consist of only three real numbers $-\frac{1}{\sqrt{\alpha}}$, 0, and $\frac{1}{\sqrt{\alpha}}$. Then, each codeword in $\mathcal{C}_\alpha$ has exactly α nonzero entries.*

In order to handle codewords for our purpose, we develop two algorithms, called Decode and USample. The Decode algorithm takes an arbitrary unit vector in S^{n-1} as an input and returns a codeword that is the closest to the input vector. The USample algorithm efficiently samples a codeword with uniform distribution over $\mathcal{C}_\alpha$. The detailed descriptions of both algorithms are presented in Algorithms 3 and 4, respectively.

We present a useful proposition showing features of the proposed ECC $\mathcal{C}_\alpha$ in terms of our requirements.

Proposition 1. *Let $\mathcal{C}_\alpha$ be the set of codewords defined in Definition 1.*

1. *The minimum angular distance between any two distinct codewords in $\mathcal{C}_\alpha$ is $\cos^{-1}(1 - \frac{1}{\alpha})$.*
2. *The output of the USample algorithm is uniformly distributed over $\mathcal{C}_\alpha$.*
3. *For any $\mathbf{u} \in S^{n-1}$, the output $\mathbf{c}$ of the Decode algorithm satisfies the following inequality:*

$$\langle \mathbf{c}, \mathbf{u} \rangle \geq \langle \mathbf{c}', \mathbf{u} \rangle \text{ for all } \mathbf{c}' \in \mathcal{C}_\alpha.$$

4. *The number of all codewords in $\mathcal{C}_\alpha$ is $\binom{n}{\alpha} \cdot 2^\alpha$.*

Algorithm 3: Decode Algorithm

Input : $\mathbf{u} = (u_1, \ldots, u_n) \in S^{n-1}$
Output: $\mathbf{c} = (c_1, \ldots, c_n) \in S^{n-1}$ such that
$\langle \mathbf{c}, \mathbf{u} \rangle \geq \langle \mathbf{c}', \mathbf{u} \rangle$ for all $\mathbf{c}' \in \mathcal{C}_\alpha$
1 Find a set of indices $J \subset [1, n]$ of size α such that
$\forall j \in J$ and $\forall k \in [1, n] \setminus J, |u_j| \geq |u_k|$.
2 **if** $j \in J$ **then** $c_j \leftarrow \frac{u_j}{|u_j|\sqrt{\alpha}}$
else $c_j \leftarrow 0$
3 Output $\mathbf{c} := (c_1, \ldots, c_n)$.

Algorithm 4: USample Algorithm

Input : Randomness seed
Output: $\mathbf{c} = (c_1, \ldots, c_n) \in S^{n-1}$ that is uniformly
sampled from $\mathcal{C}_\alpha$
1 Choose α distinct positions $j_1, \ldots, j_\alpha$ from
$\{1, \ldots, n\}$ and set $J := \{j_1, \ldots, j_\alpha\}$.
2 For $j \in J$, set c_j to one of $-\frac{1}{\sqrt{\alpha}}$ and $\frac{1}{\sqrt{\alpha}}$ at random.
For other indices $j \notin J$ set $c_j := 0$.
3 Output $\mathbf{c} := (c_1, \ldots, c_n)$.

Due to the space constraint, the proof of Proposition 1 is relegated to Appendix B of the full version [16].

The above proposition tells us that the proposed ECC, $\mathcal{C}_\alpha$, is a suitable candidate for ECCs in IronMask. The first result gives the minimum angular distance of $\mathcal{C}_\alpha$ and it confirms that $\mathcal{C}_\alpha$ is fairly *discriminative*. However, it is still unclear, as of yet, whether such the minimum distance is sufficient for real world applications. To clarify it, we will complement this argument both experimentally (in Section 4) and theoretically (in Appendix B of the full version [16]). The second and third results show that $\mathcal{C}_\alpha$ is *uniformly-sampleable* and *efficiently-decodable*, respectively. In practical face recognition systems such as ArcFace [5] and CosFace [37], n is set to be 512. Thus, if α is set appropriately, the fourth result shows that $\mathcal{C}_\alpha$ satisfies the last requirement. For example, the number of all codewords in $\mathcal{C}_{16}$ with $n = 512$ is larger than 2^{115}.

2.2. Hidden Rotation Matrix Generation

In this subsection, we construct an HRMG algorithm that takes two input vectors $\mathbf{t}, \mathbf{c}$ in S^{n-1} and returns an isometry which preserves angular distance (equivalently, inner product) as well as maps from $\mathbf{t}$ to $\mathbf{c}$.

Naive Approach. First, we present a naive approach to generate such an isometry, a rotation in the plane uniquely defined by two input vectors $\mathbf{t}$ and $\mathbf{c}$. More precisely, let
$\mathbf{w} = \mathbf{c} - \mathbf{t}^T \mathbf{c} \mathbf{t}$ and $\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ where
$\theta = \cos^{-1}(\langle \mathbf{t}, \mathbf{c} \rangle)$. Then, the naive isometry $\mathbf{R}$ mapping

from $\mathbf{t}$ to $\mathbf{c}$ can be precisely calculated as follows:

$$\mathbf{R} = \underbrace{\mathbf{I} - \mathbf{t}\mathbf{t}^T - \mathbf{w}\mathbf{w}^T}_{\text{projection part}} + \underbrace{[\mathbf{t}\ \mathbf{w}]\mathbf{R}_\theta[\mathbf{t}\ \mathbf{w}]^T}_{\text{rotation part}}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix.

Proposition 2. Given $\mathbf{t}$ and $\mathbf{c}$, *the above matrix $\mathbf{R}$ is an orthogonal matrix such that $\mathbf{Rt} = \mathbf{c}$.*

The proof of Proposition 2 is given in Appendix B of the full version [16], due to the space limitation.

Template-Protecting Approach. Although the above naive algorithm functions correctly, from the privacy point of view it might leak some information on input vectors $\mathbf{t}$ and $\mathbf{c}$ because it only rotates on the plane that includes both $\mathbf{t}$ and $\mathbf{c}$ which should be kept secret in our application, biometric recognition system. As aforementioned, we expect that the algorithm $\text{HRMG}(\mathbf{t}, \mathbf{c})$ hides information on $\mathbf{t}$ as much as possible. To this end, we add a randomizing step to blind two input vectors while preserving the necessary property for mapping from $\mathbf{t}$ to $\mathbf{c}$. The proposed HRMG algorithm is described in Algorithm 5 and a pictorial explanation for the HRMG algorithm is given in Figure 2.

Algorithm 5: HRMG Algorithm

Input : A pair of template and codeword $(\mathbf{t}, \mathbf{c})$
Output: An isometry $\mathbf{P} \in O(n)$ mapping $\mathbf{t} \mapsto \mathbf{c}$
1 Choose an orthogonal matrix $\mathbf{Q} \in O(n)$ at random.
2 Compute a rotation matrix $\mathbf{R}$ mapping $\mathbf{Q}\mathbf{t} \mapsto \mathbf{c}$.
3 Output $\mathbf{P} := \mathbf{RQ}$.

As for the first step of the HRMG Algorithm, we use a well-known algorithm [29] which generates a random orthogonal matrix $\mathbf{Q}$ from the uniform distribution. As for the second step to generate $\mathbf{R}$, we use a naive rotation matrix generation given at the beginning of this subsection.
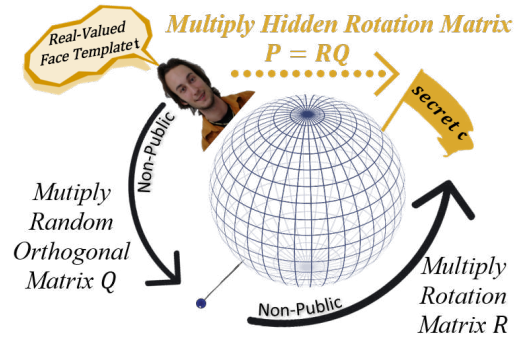

Figure 2: Geometrical interpretation of an orthogonal matrix $\mathbf{Q}$, a rotation matrix $\mathbf{R}$, and the final result $\mathbf{P}$

3. Security Analysis

In this section, we look into the security of IronMask, proposed in Section 2. For this purpose, we first introduce security requirements for biometric recognition system, and then present plausible attack scenarios and their counter-measures.

3.1. Security Requirements

For the security of biometric recognition systems, the following three conditions are required to be satisfied.

- Irreversibility: It is computationally infeasible to recover the original biometric data from the protected template.
- Revocability: It is possible to issue new protected templates to replace the compromised one.
- Unlinkability: It is computationally infeasible to retrieve any information from two protected templates generated in two different applications.

In IronMask, the Protect algorithm outputs a pair of a hash output of a randomly selected codeword and an orthogonal matrix which maps from a biometric template to the selected codeword. So, we assume that the adversary can access to a database that stores those pairs, which were generated by the Protect algorithm.

3.2. Plausible Attacks and Their Countermeasures

Now, we look into irreversibility, revocability, and unlinkability of IronMask by describing several plausible attacks and then presenting countermeasures.

Irreversibility. Given a pair of $(\mathrm{r}, \mathbf{P})$, the goal of the adversary who wants to break the irreversibility of IronMask, is to recover the biometric template $\mathbf{t}$, which was the input of the Protect algorithm once the given $(\mathrm{r}, \mathbf{P})$ was generated. Then, they hold that $\mathbf{c} = \mathbf{Pt}$ and $\mathrm{r} = \mathsf{H}(\mathbf{c})$ for the exploited hash function H. Since $\mathbf{P}$ is an invertible matrix, there exists a unique $\mathbf{t}$ for each $\mathbf{c}$ in the set of codewords, $\mathcal{C}_\alpha$. However, if the distribution of $\mathbf{t}$ is hidden, the only way to check whether $(\mathbf{t}, \mathbf{c})$ is the solution to $\mathbf{c} = \mathbf{Pt}$ and $\mathrm{r} = \mathsf{H}(\mathbf{c})$, is to check whether r is $\mathsf{H}(\mathbf{c})$ or not. On the other hand, there is no known result about the distribution of biometric templates generated by the face recognition systems that we consider in this paper. So, it is reasonable to assume that the distribution of $\mathbf{t}$ is hidden to the adversary. Therefore, under assuming that the distribution of $\mathbf{t}$ is hidden, the exploited hash function H is one-way, and the cardinality of $\mathcal{C}_\alpha$ is sufficiently large, e.g., larger than 2^{80}, it is computationally infeasible to recover $\mathbf{t}$ from the given $(\mathrm{r}, \mathbf{P})$.

Revocability. For the revocability, it should be guaranteed that the irreversibility holds with a newly issued protected template, even though the old protected templates generated by the same biometric source are given to the adversary. In our setting, since the underlying face recognition systems are based on CNN, once a user would like to generate a new protected template again for the revocation, the unprotected template $\mathbf{t}$ is newly generated. It is different from the previously generated ones, but the distances between them are very close. By considering this feature, the revocability can be regarded that any $\mathbf{t}_i$'s should not be revealed once several $(\mathrm{r}_i, \mathbf{P}_i)$'s are given, where $(\mathrm{r}_i, \mathbf{P}_i)$ is the outcome of the Protect algorithm with input $\mathbf{t}_i$ and each $\mathbf{t}_i$ is generated from the same biometric source.

Because IronMask satisfies the irreversibility, it is computationally infeasible to recover $\mathbf{t}_i$ from given a pair of $(\mathrm{r}_i, \mathbf{P}_i)$ only. However, by using several pairs of $(\mathrm{r}_i, \mathbf{P}_i)$, the adversary may obtain relations $\mathbf{P}_i^T \mathbf{c}_i = \mathbf{P}_j^T \mathbf{c}_j + \mathbf{e}_{ij}$ for each i, j where $\mathbf{e}_{ij} = \mathbf{t}_i - \mathbf{t}_j$ since the relations $\mathbf{c}_i = \mathbf{P}_i \mathbf{t}_i$, $\mathbf{c}_j = \mathbf{P}_j \mathbf{t}_j$ hold and $\mathbf{P}_i$'s are orthogonal matrices. It is equivalent to $\mathbf{c}_i = \mathbf{P}_i \mathbf{P}_j^T \mathbf{c}_j + \mathbf{P}_i \mathbf{e}_{ij}$ and thus we replace breaking the revocability by finding $(\mathbf{c}_i, \mathbf{c}_j, \mathbf{e}_{ij})$ such that

$$\mathbf{Pc}_j + \mathbf{P}_i \mathbf{e}_{ij} = \mathbf{c}_i \text{ with } \mathbf{e}_{ij} = \mathbf{t}_i - \mathbf{t}_j,$$
$$\mathbf{P}_i \mathbf{t}_i = \mathbf{c}_i \in \mathcal{C}_\alpha, \text{ and } \mathbf{P}_j \mathbf{t}_j = \mathbf{c}_j \in \mathcal{C}_\alpha \quad (1)$$

for given orthogonal matrices $\mathbf{P}_i$, $\mathbf{P}_j$, and $\mathbf{P} = \mathbf{P}_i \mathbf{P}_j^T$. Here, we again remark that $\mathbf{c}_i, \mathbf{c}_j$ are codewords and so they are sparse unit vectors of a fixed small weight whose nonzero elements have the same absolute value.

A naive way to resolve the above problem is to check all possible candidates, but it is computationally infeasible if the cardinality of the set of codewords is exponentially large in the security parameter. Someone who is familiar with mathematical structures may consider to apply structural attacks, such as lattice reductions. However, unfortunately, since $\mathbf{P}_i$'s (and thus $\mathbf{P}$) are orthogonal matrices, a lattice reduction with input $\mathbf{P}_i$ returns $\mathbf{P}_i$ itself.

Also, someone who is familiar with cryptanalysis may consider to apply a time-memory-trade-off (TMTO) strategy whose complexity is approximately a square-root of the cardinality of the search space. However, $\mathbf{e}_{ij}$ is also unknown and its space is very huge since it is a vector of small real numbers. Thus, the search space for $(\mathbf{c}_i, \mathbf{c}_j, \mathbf{e}_{ij})$ is sufficiently large under appropriate parameters and this type of attacks does not work well. We elaborate the details of the TMTO strategy to solve Equation (1) in Appendix C of the full version [16].

Unlinkability. For the unlinkability, the adversary should not find any information of the biometric data from two protected templates generated in two different applications. Suppose that two protected templates $(\mathrm{r}_1, \mathbf{P}_1)$ and $(\mathrm{r}_2, \mathbf{P}_2)$ are given to the adversary. If they are generated from the same biometric source, it has a solution to Equation (1), as in the case of revocability. That is, the adversary knows that

there is a solution to Equation (1) if he can break the unlinkability of IronMask. To the best of our knowledge, however, the best way to decide the existence of such a solution to Equation (1), is to find it directly. It is infeasible under our parameter setting, as in the case of revocability. Thus, IronMask satisfies the unlinkability.

4. Performance Evaluation

In this section, we investigate implementation aspects of IronMask combining with main CNN-based face recognition systems.

4.1. Face Datasets and Recognition Algorithm

We use three popular datasets targeted face recognition to demonstrate the feasibility of IronMask. Below we briefly explain which face images were used in each dataset for our experiments.

- CMU Multi-PIE Dataset [28]: It consists of 750,000 face images of 337 subjects. The photos are captured under 4 different sessions, 15 different rotated view points, and 19 illumination conditions. We selected 20 face images randomly in the sessions 3 and 4 for 194 common subjects each. Then, we used 5 poses (p04, p05.0, p05.1, p13, p14) and 2 illumination conditions (i08, i18) for each session, respectively.
- FEI Dataset [33]: It consists of 2,800 face images of 200 subjects. The photos are captured under 14 different rotated view points for each subject. We selected 11 poses (p02, p03, p04, p05, p06, p07, p08, p09, p11, p12, p13) among them for each subject in our experiments.
- Color-FERET Dataset [24]: We used 4 face images for 237 subjects each where the photos are varied with respect to pose, illumination, expression, and glasses.

In order to show compatibility of our proposed modular architecture, we implement it over two notable face recognition systems using angular distance metric. One of them is CosFace in [37] and the other is ArcFace in [5]. We note that ArcFace that achieves the state-of-the-art performance on the NIST Face Recognition Vendor Test (FRVT)[1]. In our experiments, we use ResNet50 with 512-dimensional output as a backbone scheme for both ArcFace and CosFace.

In many real world applications, multiple face images for a single user are available during the enrollment, so that multiple templates are also available once applying face recognition. As an intermediate process before applying our IronMask, we find a kind of center point of multiple face templates and use it as the final template of user, where the center point has a minimal cost in inner prod-

ucts with multiple templates.[2] This procedure contributes to both high compactness for intra-class and small degradation of discrimination for inter-class. We experimentally show that IronMask is well compatible with not only the original face recognition systems but also the above intermediate process with multiple images. Multiple linear regression (MLR) is sufficient for realizing the above intermediate process to find the center because the center is linearly defined from multiple templates. In our implementation, MLR is fine-tuned, where weights are normally initialized and the cost function with L1 loss is optimized using stochastic gradient descent (SGD) with learning rate (0.01, 0.001) during (1000, 500) epochs. We normalize the output of MLR and then use it as the final template, which is an input of the Protect algorithm.

4.2. Experimental Results

We provide our experimental results of IronMask over ArcFace and CosFace with the aforementioned three datasets. We consider the single-image setting and the multi-image setting, where the latter applies the intermediate process that uses a center point and the former does not. For the single-image setting, we use two standard metrics, a true accept rate (TAR) and a false accept rate (FAR), to evaluate matching performance of IronMask. For the multi-image setting, only a limited number of face images per each user is available in the target databases. Thus, in order to reasonably evaluate matching performance of Iron-Mask, we carefully design experiments as follows: 1) We randomly choose n_1 face images among n_0 total face images for each user to extract a centered input feature of the Protect algorithm. 2) We run the Verify algorithm with $n_2 (= n_0 - n_1)$ face images of the same user for TAR and $n_3 (= n_0 \times (k - 1))$ face images of different users for FAR, where k is the number of all users in each dataset. 3) We run Steps 1) and 2) r times repeatedly to get more reliable performance results. That is, we have $n_2 \times r$ TAR and $n_3 \times r$ FAR tests. For the parameter $(k, n_0, n_1, n_2, n_3, r)$, we choose $(194, 20, 5, 15, 20 \times 193, 10)$ for the CMU Multi-PIE dataset, $(200, 11, 5, 6, 11 \times 199, 10)$ for the FEI dataset, and $(237, 4, 2, 2, 4 \times 238, 6)$ for the Color-FERET dataset according to Section 4.1.

The graphs of Figure 3 present our experimental results and Table 1 summarizes TARs/FARs with respect to face recognition system and dataset. Our construction with Arc-Face (resp. CosFace) achieves 99.79% (resp. 95.78%) TAR at 0.0005% (resp. 0%) FAR when using the proposed ECC $\mathcal{C}_{16}$ with $n = 512$ (resp. $n = 512$) that provides at least 115 bits of security level. We further improve the match-

Recog	CMU Multi-PIE	FEI	Color-FERET
AF	99.05@0	99.63@0	98.13@0
AF+I	98.95@0	99.27@0	98.06@0
AF+C	99.57@9e-4	99.82@0	99.78@0
AF+I+C	99.96@1.3e-3	99.96@3e-4	99.46@0
CF	96.16@6e-4	98.69@0	96.55@0
CF+I	75.56@0	87.52@0	80.32@0
CF+C	99.92@1.3e-3	99.92@0	98.96@2e-4
CF+I+C	97.37@0	98.53@0	91.45@0

Table 1: TAR@FAR performance of 8 face recognitions for three databases are given. 'AF' and 'CF' indicate ArcFace and CosFace, respectively, and '+C' indicates whether applying the intermediate process using multiple face images and '+I' indicates whether applying the proposed template protection method, IronMask.
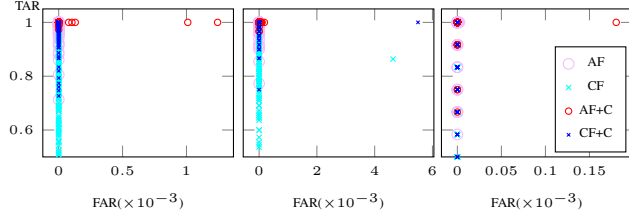

Figure 3: The experimental results on TARs at different FARs of our construction for each dataset and face recognition algorithm. The datasets used for graphs from left to right are CMU Multi-PIE, FEI, and Color-FERET each.

ing performance by using centered templates as inputs of the Protect algorithm, as described in Section 4.1. For instance, it provides about 1% (resp. 22%) improvement in TAR at the cost of 0.0013% (resp. 0%) degradation in FAR for the CMU Multi-PIE dataset with ArcFace (resp. CosFace).

One may have doubts that the relatively short minimum distance of the proposed ECC $\mathcal{C}_\alpha$ causes serious performance degradation of IronMask. According to Proposition 1, the minimum distance of $\mathcal{C}_{16}$ is $\cos^{-1}(\frac{15}{16})$, which is about $20.36°$, and so the theoretical error capacity is about $10.18° = 20.36°/2$. In case of ArcFace, the threshold which decides the closeness between two templates is about $37°$ and thus the gap between ArcFace and ours looks quite large. However, since there are only a limited number of pairs of codewords whose distance matches the minimum and most pairs of codewords have a longer distance than the minimum in $\mathcal{C}_{16}$, our Decode algorithm may decode a longer distance than the minimum distance for many cases. We confirm the above feature by comparing the performance between the original face recognition systems and IronMask combined with them. In Table 1, IronMask with ArcFace (resp. CosFace) degrades a TAR of 0.18 % (resp. 16.00 %) in average at the almost same FAR.

We also compare IronMask over ArcFace with the recent deep neural network based template protection methods. Table 2 presents comparison results of ours with the state-of-

Dataset	Alg	ET	OT	TAR@FAR
CMU Multi-PIE	[31]	Z	B255	81.40@1e-2
			B1023	81.20@1e-2
	Ours	Z	R512	98.95@0
		Z(C)	R512	99.96@1.3e-3
FEI	[13]	O	B256	99.73@0
			B1024	99.85@0
		M	B256	99.84@0
			B1024	99.98@0
	Ours	Z	R512	99.27@0
		Z(C)	R512	99.96@3e-4
Color-FERET	[13]	O	B256	98.31@0
			B1024	99.13@0
		M	B256	98.69@0
			B1024	99.24@0
	Ours	Z	R512	98.06@0
		Z(C)	R512	99.46@0

Table 2: The performance comparison among CNN-based template protected face recognition systems is given. In the ET (enrollment type) column, 'Z', 'O', and 'M' mean that the number of face images required for training during the enrollment are zero, one, and multiple, respectively. That is, both 'O' and 'M' belong to the training-in-enrollment category, but 'Z' belongs to the training-before-enrollment category. 'Ours/Z' indicates the combination of ArcFace (ResNet50) and IronMask. 'Ours/Z(C)' indicates the combination of ArcFace (ResNet50), the center finding process (C), and IronMask. In the OT (output type) column, 'B' means binary and 'R' means real.

the-art researches for each dataset: [31] for CMU Multi-PIE dataset and [13] for FEI and Color-FERET datasets. The main difference between ours and others is that we use a real-valued face feature vector itself without binarization, which may cause significant deterioration in the matching performance. In Table 2, ours improves a TAR of at least 17.7% than [31] with the zero-shot enrollment. We note that the zero-shot enrollment does not re-train even if a new identity is enrolled in the authentication system. Table 2 also shows that ours has comparable matching performance to the schemes that re-train using one or multiple images in the enrollment system.

For the readers who are interested in time and storage required for running IronMask, we report our experimental results in terms of running times and storage below. We have implemented our construction on Linux using a single machine with Intel Core i7-6820 HQ running at 2.70 GHz and 16 GB RAM. Our experiments take 250 ms for the Protect algorithm at the Linux machine. For the Verify algorithm, it takes 6.72 ms at the Linux machine. On the other hand, since the protected template $\mathbf{pt} = (r, \mathbf{P})$ consists of a hash value of codeword and an $n \times n$ real-valued matrix, the size of the protected template under $\mathcal{C}_{16}$ with $n = 512$ is less than 1 Mb for face recognition. It shows that IronMask is sufficient to be deployed in real world applications such as WLogin services using biometric authentication.

References

[1] M Ao and S Z Li. Near infrared face based biometric key binding. In *International Confernece on Biometrics*, pages 376–385. Springer, 2009. 3

[2] Vishnu Naresh Boddeti. Secure face matching using fully homomorphic encryption. *BTAS*, pages 1–10, 2018. 3

[3] Kai Cao and Anil K Jain. Learning fingerprint reconstruction: From minutiae to image. *IEEE Transactions on information forensics and security*, 10(1):104–117, 2015. 1

[4] R Cappelli, A Lumini, D Maio, and D Maltoni. Fingerprint image reconstruction from standard templates. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 29, pages 1489–1503. arXiv preprint, 2007. 1

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 2, 3, 5, 7

[6] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *International conference on the theory and applications of cryptographic techniques*, pages 523–540. Springer, 2004. 3

[7] Joshua J. Engelsma, Anil K. Jain, and Vishnu Naresh Boddeti. Hers: Homomorphically encrypted representation search. *arXiv:2003.12197*, 2020. 3

[8] Yi C Feng, Pong C Yuen, and Anil K Jain. A hybrid approach for generating secure and discriminating face template. *IEEE transactions on information forensics and security*, 5(1):103–117, 2009. 3

[9] M Fredrikson, S Jha, and T Ristenpart. Model inversion attacks that exploit confidence information ad basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015. 1

[10] Javier Galbally, Arun Ross, Marta Gomez-Barrero, Julian Fierrez, and Javier Ortega-Garcia. Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms. *Computer Vision and Image Understanding*, 117(10):1512–1525, 2013. 1

[11] Siddharth Gopal and Yiming Yang. Von mises-fisher clustering models. In *International Conference on Machine Learning*, pages 154–162, 2014. 2

[12] Arun Kumar Jindal, Srinivasa Rao Chalamala, and Santosh Kumar Jami. Face template protection using deep convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 462–470, 2018. 1, 3

[13] Arun Kumar Jindal, Srinivasa Rao Chalamala, and Santosh Kumar Jami. Securing face templates using deep convolutional neural network and random projection. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6. IEEE, 2019. 1, 3, 8

[14] Ari Juels and Madhu Sudan. A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security*, pages 28–36, 1999. 3

[15] Ari Juels and Madhu Sudan. A fuzzy vault scheme. In *Designs, Codes and Cryptography*, volume 38, pages 237–257, 2006. 3

[16] Sunpill Kim, Yunseong Jeong, Jinsu Kim, Jungkon Kim, Hyung Tae Lee, and Jae Hong Seo. IronMask: Modular architecture for protecting deep face template. *arXiv preprint*, 2021. 3, 5, 6

[17] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 2

[18] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *International Conference on Machine Learning*, pages 507—-516, 2016. 2

[19] Yu Liu, Hongyang Li, and Xiaogang Wang. Rethinking feature discrimination and polymerization for large-scale recognition. In *arXiv:1710.00870*. arXiv preprint, 2017. 2

[20] Guangcan Mai, Kai Cao, Pong C Yuen, and Anil K Jain. On the reconstruction of face images from deep face templates. *IEEE transactions on pattern analysis and machine intelligence*, 41(5):1188–1202, 2018. 1

[21] Deen Dayal Mohan, Nishant Sankaran, and Sergey Tulyakov. Significant feature based representation for template protection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 1, 2, 3

[22] Rohit Kumar Pandey and Venu Govindaraju. Secure face template generation via local region hashing. In *International Conference on Biometrics*, pages 299–304. IEEE, 2015. 3

[23] Rohit Kumar Pandey, Yingbo Zhou, Bhargava Urala Kota, and Venu Govindaraju. Deep secure encoding for face template protection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 77–83. IEEE, 2016. 1, 3

[24] P. Jonathon Phillips, Harry Wechsler, Jeffrey Huang, and Patrick J. Rauss. The FERET database and evaluation procedure for face-recognition algorithms. *Image Vision Computing*, 16(5):295–306, 1998. 2, 7

[25] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. In *arXiv:1703.09507*. arXiv preprint, 2017. 2

[26] N K Ratha, S Chikkerur, J H Connell, and R M Bolle. Generating cancelable fingerprint templates. *IEEE transactions on pattern analysis and machine intelligence*, 29(4):561–572, 2007. 3

[27] F Schroff, D Kalenichenko, and J Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 3

[28] Terence Sim, Simon Baker, and Maan Bsat. The CMU pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003. 2, 7

[29] Gilbert W Stewart. The efficient generation of random orthogonal matrices with an application to condition estima-

tors. *SIAM Journal on Numerical Analysis*, 17(3):403–409, 1980. 5

[30] Y Sutcu, Q Li, and N Memon. Protecting biometric tempaltes with sketch: Theory and practice. In *IEEE transactions on Information and Forensics Security*, volume 2, pages 503–512. IEEE, 2007. 3

[31] Veeru Talreja, Matthew C Valenti, and Nasser M Nasrabadi. Zero-shot deep hashing and neural network based error correction for face template protection. *arXiv preprint arXiv:1908.02706*, 2019. 1, 2, 3, 8

[32] A B J Teoh, A Goh, and D C L Ngo. Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):1892–1901, 2006. 3

[33] Carlos Eduardo Thomaz and Gilson Antonio Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28(6):902–913, 2010. 2, 7

[34] S Tulyakov, F Farooq, P Mansukhani, and V Govindaraju. Symmetric hash functions for secure fingerprint biometric systems. In *Pattern Recogn. Lett.*, volume 28, pages 2472–2436, 2007. 3

[35] Feng Wang, Weiyang Liu, Haijun Liu, and Jian Cheng. Additive margin softmax for face verification. In *arXiv:1801.05599*. arXiv preprint, 2018. 2

[36] Feng Wang, Xiang Xiang, Jian Cheng, and Alan L Yuille. Normface:l2 hypersphere embedding for face verification. In *arXiv:1704.06369*. arXiv preprint, 2017. 2

[37] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. 2, 5, 7

[38] Y Wu and B Qiu. Transforming a pattern identifier into biometric key generators. In *IEEE International Conference on Multimedia and Expo*, pages 78–82. IEEE, 2010. 3

[39] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10823–10832, 2019. 2